



UNITED STATES PATENT AND TRADEMARK OFFICE

DA
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/766,336	01/28/2004	Shyh-Mei F. Ho	SVL920030102US1	3271
45112	7590	06/27/2007	EXAMINER	
Kunzler & McKenzie 8 EAST BROADWAY SUITE 600 SALT LAKE CITY, UT 84111			ALVESTEFFER, STEPHEN D	
		ART UNIT	PAPER NUMBER	
		2173		
		MAIL DATE	DELIVERY MODE	
		06/27/2007	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/766,336	HO ET AL.	
	Examiner	Art Unit	
	Stephen Alvesteffer	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 28 January 2004.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-40 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-40 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 28 January 2004 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>20040128</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application
	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

Claims 1-40 are presented for examination. Claims 1, 12, 20, 29, 39, and 40 are independent claims. The Information Disclosure Statement received on January 28, 2004 has been considered by the examiner, except for the reference represented by A5, because no publication date is provided.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1 and 11 are rejected under 35 U.S.C. 102(b) as being anticipated by Bach et al. (hereinafter Bach '739), United States Patent number 5,781,739.

Regarding claim 1, Bach '739 teaches an apparatus for automatically generating a web interface for an MFS-based IMS application (see column 1 lines 55-57; “*...translate MFS source for the purpose of running an IMS transaction*”), comprising: an import module configured to import MFS-based IMS source files corresponding to an MFS-based IMS application (see column 2 lines 62-66; “*...browse and download MFS source information*”); a metadata generator configured to store a

standardized metadata description of the Message Input Description (MID) and Message Output Description (MOD) for the MFS-based IMS application (see column 4 lines 51-59; “*IMS Web uses only the information in MFS Message Input Descriptors (MIDs-111) and MFS Message Output Descriptors (MODs-112) to format input and output messages*”); and a code generator configured to generate a middleware application corresponding to the MFS-based IMS application from the standardized metadata description, the middleware application interfacing between a client application and the corresponding MFS-based IMS application (see column 10 lines 44-49; “*The generated CGI-BIN program invokes this class to parse the input string from the Web browser*”, code is generated to interface between a web browser and the IMS database).

Regarding claim 11, Bach ‘739 teaches a deployment module configured to store the standardized metadata descriptions and middleware application in one or more repositories (see paragraph [0014]; “*Places the CGI-BIN executable file in the directory which was user specified in IMS Web Studio ('Web Server's CGI Path'). This directory must be the same directory that their Web server uses to look for CGI-BIN scripts*”, the CGI-BIN directory is the repository that stores the generated middleware applications to interface with the MFS-based IMS applications).

Claims 1, 2, 20, 29, 39, and 40 are rejected under 35 U.S.C. 102(e) as being anticipated by Chiang et al. (hereinafter Chiang), United States Patent Application Publication number 2004/0054969.

Regarding claim 1, Chiang teaches an apparatus for automatically generating a web interface for an MFS-based IMS application (see Chiang claim 1; “*computer program for generating MFS WSDL files from IMS MFS source files*”, where WSDL files are Web Service Description Language files for defining web services and can be used to define a web interface), comprising: an import module configured to import MFS-based IMS source files corresponding to an MFS-based IMS application (see paragraph [0010]; “*method for accessing MFS-based IMS applications*”); a metadata generator configured to store a standardized metadata description of the Message Input Description (MID) and Message Output Description (MOD) for the MFS-based IMS application (see paragraph [0004]; “*The MFS language utility compiles MFS source, generates MFS control blocks in a proprietary format, known as Message Input/Output Descriptors (MID/MOD), and places them in an IMS format library*”); and a code generator configured to generate a middleware application corresponding to the MFS-based IMS application from the standardized metadata description, the middleware application interfacing between a client application and the corresponding MFS-based IMS application (see paragraph [0030]; “*generic servlet is responsible for processing the HTTP XML request, invoking the adapter, and loading the stylesheet... client is able to page through logical pages and physical pages without making extra requests to the MFS XML adapter 116 (and the IMS transaction system 130)... an instance servlet is only generated for each initial MOD*”).

Regarding claim 2, Chiang teaches that the standardized metadata description comprises an eXtended Markup Language Metadata Interchange (XMI) file (see

paragraph [0023]; “*The MFS mapper generates three XMI files for the three external reference pointers. These three files include a “midname.xmi” file for each MID with its associated device input format (DIF), a “modname.xmi” file for each MOD with its associated device output format (DOF)*”).

Regarding claim 20, Chiang teaches a utility for automatically generating a web interface for an MFS-based IMS application, comprising: an import module configured to import MFS-based IMS source files corresponding to an MFS-based IMS application (see paragraph [0010]; “*method for accessing MFS-based IMS applications*”); a parser configured to parse each of the MFS-based IMS source files into one or more Message Input Description (MIDs) and one or more Message Output Description (MODs) (see paragraph [0004]; “*The MFS language utility compiles MFS source, generates MFS control blocks in a proprietary format, known as Message Input/Output Descriptors (MID/MOD), and places them in an IMS format library*”, the source files must inherently be parsed in order to produce the MID/MOD descriptors for them); a metadata generator configured to store at least one eXtended Markup Language Metadata Interchange (XMI) file for the MIDs and MODs of the MFS-based IMS application (see paragraph [0004]; “*The MFS language utility compiles MFS source, generates MFS control blocks in a proprietary format, known as Message Input/Output Descriptors (MID/MOD), and places them in an IMS format library*”); a code generator configured to generate a middleware application corresponding to the MFS-based IMS application from the standardized metadata description, the middleware application interfacing between a client application and the corresponding MFS-based IMS application (see

paragraph [0030]; "*generic servlet is responsible for processing the HTTP XML request, invoking the adapter, and loading the stylesheet... client is able to page through logical pages and physical pages without making extra requests to the MFS XML adapter 116 (and the IMS transaction system 130)... an instance servlet is only generated for each initial MOD*"; and a deployment module configured to deploy the XMI files and middleware application to one or more servers configured to enable communication between the client application and the MFS-based IMS application (see paragraphs [0022] and [0024]; "*The MFS mapper reads and parses MFS source files for a particular application and generates XMI files that describe the MFS-based application interface*" and "*the MFS XML adapter has access to an XML source repository and can properly invoke an MFS-based IMS application*", XMI files are deployed to a repository where they are retrieved by an MFS XML adapter).

Claim 29 recites a method with substantially the same limitations as the utility of claim 20. Therefore, claim 29 is rejected under the same rationale.

Claim 39 recites an apparatus with substantially the same limitations as the utility of claim 20. Therefore, claim 39 is rejected under the same rationale.

Claim 40 recites an article of manufacture with substantially the same limitations as the utility of claim 20. Therefore, claim 40 is rejected under the same rationale.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

Art Unit: 2173

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 3-6, 12-15, 21-24, and 30-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chiang (2004/0054969) *supra* and Bach et al. (hereinafter Bach '660), United States Patent number 6,141,660.

Regarding claim 3, Chiang teaches all the limitations of claim 3 except for a command-line interface configured to execute the import module, the metadata generator, and the code generator in response to a parameter set provided as a single input to the command-line interface. However, running applications from a command-line interface was well known in the art at the time the invention was made, as evidenced by Bach '660 (see column 17 lines 4-12; "*To begin using the command-line interface, the user goes to the directory in which the product was installed, and enters the command "IOCCLI." A command prompt, such as "COMMAND >>", is then displayed on the monitor and the user can begin entering CDT 400 commands*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to perform the operations as described by Chiang on a command line interface as described by Bach '660 for the purpose of providing a user interface.

Regarding claim 4, Bach '660 teaches a loader configured to load a script comprising the parameter set from persistent storage (see column 17 lines 4-12; "*the user can enter commands one by one, or can run a command script that contains all of the commands*").

Regarding claim 5, Bach '660 teaches that the script comprises a plurality of parameter sets each associated with a different MFS-based IMS application (see column 17 lines 4-12; "*the user can enter commands one by one, or can run a command script that contains all of the commands*", the command line interface described by Bach '660 is capable of comprising a plurality of parameter sets each associated with a different MFS-based IMS application).

Regarding claim 6, Bach teaches that the parameter set is manually entered, the apparatus further comprising a storage module configured to store the manually entered parameter set for subsequent automated use (see column 17 lines 54-67; "*A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403*".)

Regarding claim 12, Chiang and Bach '660 teach an apparatus for automatically generating a web interface for an MFS-based IMS application, comprising: a metadata generator configured to generate at least one eXtended Markup Language Metadata Interchange (XMI) file that stores the Message Input Description (MID) and Message Output Description (MOD) associated with a MFS-based IMS application installed on a host (see Chiang paragraph [0023]; "*The MFS mapper generates three XMI files for the three external reference pointers. These three files include a "midname.xmi" file for each MID with its associated device input format (DIF), a "modname.xmi" file for each*

MOD with its associated device output format (DOF")}; a code generator configured to generate a middleware application for the MFS-based IMS application, the middleware application configured to interface between a client application and the corresponding MFS-based IMS application (see Chiang paragraph [0030]; "generic servlet is responsible for processing the HTTP XML request, invoking the adapter, and loading the stylesheet... client is able to page through logical pages and physical pages without making extra requests to the MFS XML adapter 116 (and the IMS transaction system 130)... an instance servlet is only generated for each initial MOD"); and a command-line interface configured to execute the metadata generator and the code generator in response to a parameter set (see Bach '660 column 17 lines 4-12; "To begin using the command-line interface, the user goes to the directory in which the product was installed, and enters the command "IOCCLI." A command prompt, such as "COMMAND >>", is then displayed on the monitor and the user can begin entering CDT 400 commands").

Regarding claim 13, Bach '660 teaches a loader configured to load a script comprising the parameter set from persistent storage (see column 17 lines 4-12; "the user can enter commands one by one, or can run a command script that contains all of the commands").

Regarding claim 14, Bach '660 teaches that the parameter set is provided as a single input to the command-line interface (see column 17 lines 4-12; "To begin using the command-line interface, the user goes to the directory in which the product was installed, and enters the command "IOCCLI." A command prompt, such as "COMMAND

>>, is then displayed on the monitor and the user can begin entering CDT 400 commands”).

Regarding claim 15, Chiang teaches that the middleware application comprises a server component and a back-end component (see paragraph [0009]; “*The adapter can reside in a server that is distanced from the client program while the MFS-based IMS application can reside in a mainframe that is distanced from the server and the client program*”).

Claims 21-24 recite a utility with substantially the same limitations as claims 3-6, respectively. Therefore, they are rejected under the same rationale.

Claims 30-33 recite a method with substantially the same limitations as the apparatus of claims 3-6, respectively. Therefore, they are rejected under the same rationale.

Claims 7, 8, 16-18, 25, 26, 34, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chiang (2004/0054969) *supra* and Bach '660 (6,141,660) *supra* and Narayan, United States Patent Application Publication number 2002/0078255.

Regarding claim 7, Chiang and Bach '660 teach all the limitations of claim 7 except that the command-line interface comprises a plurality of modes, each mode comprising a different level of user interaction. However, it was well known in the art at the time the invention was made that some command-line interfaces comprised a plurality of modes, each mode comprising a different level of user interaction. Narayan

teaches an command line interface with multiple modes of operation (see paragraph [0044]; “*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*”). It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide command line interface modes as taught by Narayan in the code generating interface as taught by Chiang and Bach ‘660 in order to provide another method of interaction with the invention.

Regarding claim 8, Narayan teaches that one mode comprises a batch mode that reads the parameter set from persistent storage (see paragraph [0044]; “*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*”).

Regarding claim 16, Narayan teaches that the command-line interface comprises a plurality of modes, each mode involving a different level of user interaction (see paragraph [0044]; “*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*”).

Regarding claim 17, Narayan teaches that one mode prompts a user for each parameter of the parameter set (see paragraph [0326]; “*presents the user with a user interface that prompts for user information when appropriate server method is invoked*”).

Regarding claim 18, Narayan teaches that the command-line interface is configured to be executed by a separate software module (see paragraph [0058]; “*The UI client has a user interface component that is either graphical or command line*”).

Claims 25 and 26 recite a utility with substantially the same limitations as claims 7 and 8, respectively. Therefore, they are rejected under the same rationale.

Claims 34 and 35 recite a method with substantially the same limitations as the apparatus of claims 7 and 8, respectively. Therefore, they are rejected under the same rationale.

Claims 9, 27, and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chiang (2004/0054969) *supra* and Dan et al. (hereinafter Dan), United States Patent number 6,560,639.

Regarding claim 9, Chiang teaches all the limitations of claim 9 except an error module configured to present an error message in response to an error condition triggered by the import module, the metadata generator, or the code generator. However, including an error module to respond to application errors was well known in the art at the time the invention was made, as evidenced by Dan (see column 5 lines 1-11; “*error report manager may report any error in intended user changes to a requested web page*”). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include an error module as described by Dan in the invention of Chiang for the purpose of handling error conditions in the application.

Claim 27 recites a utility with substantially the same limitations as claim 9. Therefore, claim 27 is rejected under the same rationale.

Claim 36 recites a method with substantially the same limitations as the apparatus of claim 9. Therefore, claim 36 is rejected under the same rationale.

Claims 10, 28, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chiang (2004/0054969) *supra* and Snover et al. (hereinafter Snover), United States Patent Application Publication number 2004/0230987.

Regarding claim 10, Chiang teaches all the limitations of claim 10 except that the import module is configured to import a plurality of MFS-based IMS source files in response to a single parameter. However, it was well known in the art at the time the invention was made that a wildcard character can be entered on many command line interfaces to specify several files using a single parameter, as indicated by Snover (see paragraph [0054]; *"the service could perform wildcard expansion on a filename entered as "A*" on the command line. Before the second pass, the Name member contains "A*" as it was entered on the command line. During the second pass, the Filename service may locate a set of files starting with A and store them in the ArrayList F"*). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the wildcard expansion technique of Snover with the web interface generator of Chiang in order to make importing several files at the command line less tedious.

Claim 28 recites a utility with substantially the same limitations as claim 10. Therefore, claim 28 is rejected under the same rationale.

Claim 37 recites a method with substantially the same limitations as the apparatus of claim 10. Therefore, claim 37 is rejected under the same rationale.

Claims 19 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chiang (2004/0054969) *supra* and Bach '660 (6,141,660) *supra* and Bach '739 (5,781,739) *supra*.

Regarding claim 19, Chiang and Bach '660 teach all the limitations of claim 19 except a deployment module configured to store the XMI files and middleware application in one or more repositories. However, Bach '739 teaches a CGI-BIN directory that acts as a repository that stores the generated middleware applications to interface with the MFS-based IMS applications (see paragraph [0014]; "*Places the CGI-BIN executable file in the directory which was user specified in IMS Web Studio ('Web Server's CGI Path'). This directory must be the same directory that their Web server uses to look for CGI-BIN scripts*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to store the middleware files generated by the methods of Chiang and Bach '660 in one or more repositories as taught by Bach '739 so the files can be reused.

Claim 38 recites a method with substantially the same limitations as the apparatus of claim 19. Therefore, claim 38 is rejected under the same rationale.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Stephen Alvesteffer whose telephone number is (571) 270-1295. The examiner can normally be reached on Monday-Friday 9:30AM-6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (571)272-4048. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Stephen Alvesteffer
Examiner
Art Unit 2173.

A
6-21-2007

Tadesse Hailu
TADESSE HAILU
PRIMARY EXAMINER

